```
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBB          BBB   AAA         AAA  SSS          RRR         RRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBBBBBBBBBBB       AAA         AAA     SSSSSSSSS    RRRRRRRRRRR       TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR         TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR         TTT        LLL
BBB          BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR         TTT        LLL
BBB          BBB   AAA         AAA           SSS  RRR      RRR      TTT        LLL
BBB          BBB   AAA         AAA           SSS  RRR      RRR      TTT        LLL
BBB          BBB   AAA         AAA           SSS  RRR      RRR      TTT        LLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB       AAA         AAA  SSSSSSSSSSS  RRR         RRR    TTT        LLLLLLLLLLLLLLL
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  EEEEEEEEEE    CCCCCCCC  HH      HH    OOOOOO
BBBBBBBB      AAAAAA      SSSSSSSS  EEEEEEEEEE    CCCCCCCC  HH      HH    OOOOOO
BB      BB   AA      AA  SS        EE            CC        HH      HH   OO      OO
BB      BB   AA      AA  SS        EE            CC        HH      HH   OO      OO
BB      BB   AA      AA  SS        EE            CC        HH      HH   OO      OO
BB      BB   AA      AA  SS        EE            CC        HH      HH   OO      OO
BBBBBBBB     AA      AA    SSSSSS  EEEEEEEE      CC        HHHHHHHHHH   OO      OO
BBBBBBBB     AA      AA    SSSSSS  EEEEEEEE      CC        HHHHHHHHHH   OO      OO
BB      BB   AAAAAAAAAA          SS EE            CC        HH      HH   OO      OO
BB      BB   AAAAAAAAAA          SS EE            CC        HH      HH   OO      OO
BB      BB   AA      AA          SS EE            CC        HH      HH   OO      OO
BB      BB   AA      AA          SS EE            CC        HH      HH   OO      OO
BBBBBBBB     AA      AA  SSSSSSSS  EEEEEEEEEE    CCCCCCCC  HH      HH    OOOOOO
BBBBBBBB     AA      AA  SSSSSSSS  EEEEEEEEEE    CCCCCCCC  HH      HH    OOOOOO

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;      1          0001  0 MODULE BAS$ECHO (                                    !
;      2          0002  0                 IDENT = '1-002'                      ! File: BASECHO.B32
;      3          0003  0                 ) =
;      4          0004  1 BEGIN
;      5          0005  1 !
;      6          0006  1 !*****************************************************************************
;      7          0007  1 !*                                                                           *
;      8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;      9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     10          0010  1 !*  ALL RIGHTS RESERVED.                                                      *
;     11          0011  1 !*                                                                           *
;     12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;     13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;     14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;     15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;     16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;     17          0017  1 !*  TRANSFERRED.                                                              *
;     18          0018  1 !*                                                                           *
;     19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;     20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;     21          0021  1 !*  CORPORATION.                                                              *
;     22          0022  1 !*                                                                           *
;     23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;     24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;     25          0025  1 !*                                                                           *
;     26          0026  1 !*                                                                           *
;     27          0027  1 !*****************************************************************************
;     28          0028  1 !
;     29          0029  1
;     30          0030  1 !++
;     31          0031  1 ! FACILITY:  BASIC-PLUS-2 Miscellaneous I/O
;     32          0032  1 !
;     33          0033  1 ! ABSTRACT:
;     34          0034  1 !
;     35          0035  1 !       This module contains the BASIC ECHO and NOECHO functions,
;     36          0036  1 !       Which turn on and off echoing on a specified channel.
;     37          0037  1 !
;     38          0038  1 ! ENVIRONMENT:  VAX-11 User Mode
;     39          0039  1 !
;     40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: 17-APR-1979
;     41          0041  1 !
;     42          0042  1 ! MODIFIED BY:
;     43          0043  1 !
;     44          0044  1 ! 1-001 - Original.
;     45          0045  1 ! 1-002 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
;     46          0046  1 !--
;     47          0047  1
;     48          0048  1 !<BLF/PAGE>
```

```
;     50          0049  1 !
;     51          0050  1 ! SWITCHES:
;     52          0051  1 !
;     53          0052  1
;     54          0053  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;     55          0054  1
;     56          0055  1 !
;     57          0056  1 ! LINKAGES:
;     58          0057  1 !
;     59          0058  1
;     60          0059  1 REQUIRE 'RTLIN:OTSLNK';                         ! Define linkages
;     61          0488  1
;     62          0489  1 !
;     63          0490  1 ! TABLE OF CONTENTS:
;     64          0491  1 !
;     65          0492  1
;     66          0493  1 FORWARD ROUTINE
;     67          0494  1     BAS$ECHO,                                   ! Enable echoing
;     68          0495  1     BAS$NOECHO;                                 ! Disable echoing
;     69          0496  1
;     70          0497  1 !
;     71          0498  1 ! INCLUDE FILES:
;     72          0499  1 !
;     73          0500  1
;     74          0501  1 REQUIRE 'RTLML:OTSLUB';                         ! Get LUB definitions
;     75          0641  1
;     76          0642  1 REQUIRE 'RTLML:OTSISB';                         ! Get ISB definitions
;     77          0810  1
;     78          0811  1 REQUIRE 'RTLIN:RTLPSECT';                       ! Macros for defining psects
;     79          0906  1
;     80          0907  1 LIBRARY 'RTLSTARLE';                            ! System symbols
;     81          0908  1
;     82          0909  1 !
;     83          0910  1 ! MACROS:
;     84          0911  1 !
;     85          0912  1 !       NONE
;     86          0913  1 !
;     87          0914  1 ! EQUATED SYMBOLS:
;     88          0915  1 !
;     89          0916  1 !       NONE
;     90          0917  1 !
;     91          0918  1 ! PSECTS:
;     92          0919  1 !
;     93          0920  1 DECLARE_PSECTS (BAS);                           ! Declare psects for BAS$ facility
;     94          0921  1 !
;     95          0922  1 ! OWN STORAGE:
;     96          0923  1 !
;     97          0924  1 !       NONE
;     98          0925  1 !
;     99          0926  1 ! EXTERNAL REFERENCES:
;    100          0927  1 !
;    101          0928  1
;    102          0929  1 EXTERNAL ROUTINE
;    103          0930  1     BAS$$OPEN_ZERO : NOVALUE,                   ! Open channel zero
;    104          0931  1     BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,         ! Load register CCB
;    105          0932  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,           ! Done with register CCB
;    106          0933  1     BAS$$STOP_IO : NOVALUE;                     ! Signal fatal I/O error
```

```
;  107          0934  1
;  108          0935  1 !+
;  109          0936  1 ! The following are the error codes used in this module.
;  110          0937  1 !-
;  111          0938  1
;  112          0939  1 EXTERNAL LITERAL
;  113          0940  1     BAS$K_IO_CHANOT : UNSIGNED (8);             ! Channel not open.
;  114          0941  1
```

```
;  116     0942  1 GLOBAL ROUTINE BAS$ECHO (                     ! Enable echoing
;  117     0943  1         CHAN                                  ! Channel on which to enable echoing
;  118     0944  1     ) =
;  119     0945  1
;  120     0946  1 !++
;  121     0947  1 ! FUNCTIONAL DESCRIPTION:
;  122     0948  1 !
;  123     0949  1 !       Enables echoing on the terminal open on the specified channel.
;  124     0950  1 !
;  125     0951  1 ! FORMAL PARAMETERS:
;  126     0952  1 !
;  127     0953  1 !       CHAN.rl.v       The channel whose terminal to enable echoing on
;  128     0954  1 !
;  129     0955  1 ! IMPLICIT INPUTS:
;  130     0956  1 !
;  131     0957  1 !       NONE
;  132     0958  1 !
;  133     0959  1 ! IMPLICIT OUTPUTS:
;  134     0960  1 !
;  135     0961  1 !       LUB$V_NOECHO which, when set, suppresses terminal echoing.
;  136     0962  1 !
;  137     0963  1 ! ROUTINE VALUE:
;  138     0964  1 ! COMPLETION CODES:
;  139     0965  1 !
;  140     0966  1 !       SS$_NORMAL
;  141     0967  1 !
;  142     0968  1 ! SIDE EFFECTS:
;  143     0969  1 !
;  144     0970  1 !       Signals if an error is encountered.
;  145     0971  1 !       BAS$$CB_PUSH will signal if the channel number is invalid.
;  146     0972  1 !       We signal BAS$K_IO_CHANOT if the channel is not open.
;  147     0973  1 !
;  148     0974  1 !--
;  149     0975  1
;  150     0976  2     BEGIN
;  151     0977  2
;  152     0978  2     BUILTIN
;  153     0979  2         FP;
;  154     0980  2
;  155     0981  2     GLOBAL REGISTER
;  156     0982  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  157     0983  2
;  158     0984  2     LOCAL
;  159     0985  2         FMP : REF BLOCK [, BYTE];
;  160     0986  2
;  161     0987  2     FMP = .FP;
;  162     0988  2 !+
;  163     0989  2 ! Get the CCB for the channel.
;  164     0990  2 !-
;  165     0991  2
;  166     0992  3     IF (.CHAN EQL 0)
;  167     0993  2     THEN
;  168     0994  3         BEGIN
;  169     0995  3 !+
;  170     0996  3 ! The user is referencing his controlling terminal.
;  171     0997  3 !-
;  172     0998  3         BAS$$CB_PUSH (LUB$K_LUN_INPU, LUB$K_ILUN_MIN);
```

```
;  173          0999  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  174          1000  3      !+
;  175          1001  3      ! If the controlling terminal is not yet open, open it.
;  176          1002  3      !-
;  177          1003  3
;  178          1004  3          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
;  179          1005  3
;  180          1006  3          END
;  181          1007  2      ELSE
;  182          1008  3          BEGIN
;  183          1009  3      !+
;  184          1010  3      ! This is an ordinary channel.
;  185          1011  3      !-
;  186          1012  3          BAS$$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
;  187          1013  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  188          1014  2          END;
;  189          1015  2
;  190          1016  2      !+
;  191          1017  2      ! If the channel is not now open, either there is a problem with
;  192          1018  2      ! the OPEN code, or the non-zero channel was not first opened.
;  193          1019  2      !-
;  194          1020  2
;  195          1021  2      IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  196          1022  2
;  197          1023  2      !+
;  198          1024  2      ! Now clear the NOECHO bit, which will cause the record level code
;  199          1025  2      ! to tell RMS to return to echoing terminal input.
;  200          1026  2      !-
;  201          1027  2      CCB [LUB$V_NOECHO] = 0;
;  202          1028  2      !+
;  203          1029  2      ! We are done with register CCB.
;  204          1030  2      !-
;  205          1031  2      BAS$$CB_POP ();
;  206          1032  2      RETURN (SS$_NORMAL);
;  207          1033  1      END;                                    ! end of BAS$ECHO

                                                                     .TITLE  BAS$ECHO
                                                                     .IDENT  \1-002\

                                                                     .EXTRN  BAS$$OPEN_ZERO, BAS$$CB_PUSH
                                                                     .EXTRN  BAS$$CB_POP, BAS$$STOP_IO
                                                                     .EXTRN  BAS$K_IO_CHANOT

                                                                     .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                                 081C 00000          .ENTRY  BAS$ECHO, Save R2,R3,R4,R11                     ; 0942
                                  54 00000000G   00   9E 00002      MOVAB   BAS$$CB_PUSH, R4                                ;
                                  53             5D   D0 00009      MOVL    FP, FMP                                         ; 0987
                                           04    AC   D5 0000C      TSTL    CHAN                                            ; 0992
                                                 1E   12 0000F      BNEQ    1$                                              ;
                                  50             08   CE 00011      MNEGL   #8, R0                                          ; 0998
                                  52             07   CE 00014      MNEGL   #7, R2                                          ;
                                                 64   16 00017      JSB     BAS$$CB_PUSH                                    ;
                              FF4C CB            0C   A3 D0 00019   MOVL    12(FMP), -180(CCB)                              ; 0999
                                  29             FC   AB E8 0001F   BLBS    -4(CCB), 3$                                     ; 1004
                                                 0C   A3 DD 00023   PUSHL   12(FMP)                                         ;
```

```
                00000000G  00          01  FB 00026          CALLS   #1, BAS$$OPEN_ZERO                       ;
                                       0E  11 0002D          BRB     2$                                      ; 0992
                                       50  D4 0002F  1$:     CLRL    R0                                      ; 1012
                           52      04  AC  D0 00031          MOVL    CHAN, R2                                ;
                                       64  16 00035          JSB     BAS$$CB_PUSH                            ;
                    FF4C   CB      0C  A3  D0 00037          MOVL    12(FMP), -180(CCB)                      ; 1013
                           0B      FC  AB  E8 0003D  2$:     BLBS    -4(CCB), 3$                             ; 1021
                           7E      00G 8F  9A 00041          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                 ;
                00000000G  00          01  FB 00045          CALLS   #1, BAS$$STOP_IO                        ;
                      A0   AB          01  8A 0004C  3$:     BICB2   #1, -96(CCB)                            ; 1027
                                00000000G  00  16 00050          JSB     BAS$$CB_POP                             ; 1031
                           50          01  D0 00056          MOVL    #1, R0                                  ; 1032
                                           04 00059          RET                                             ; 1033

; Routine Size:  90 bytes,    Routine Base:  _BAS$CODE + 0000

;  208          1034  1
```

```
;  210     1035  1 GLOBAL ROUTINE BAS$NOECHO (                          ! Disable echoing
;  211     1036  1         CHAN                                         ! Channel on which to disable echoing
;  212     1037  1     ) =
;  213     1038  1
;  214     1039  1 !++
;  215     1040  1 ! FUNCTIONAL DESCRIPTION:
;  216     1041  1 !
;  217     1042  1 !       Disables echoing on the terminal open on the specified channel.
;  218     1043  1 !
;  219     1044  1 ! FORMAL PARAMETERS:
;  220     1045  1 !
;  221     1046  1 !       CHAN.rl.v       The channel whose terminal to disable echoing on
;  222     1047  1 !
;  223     1048  1 ! IMPLICIT INPUTS:
;  224     1049  1 !
;  225     1050  1 !       NONE
;  226     1051  1 !
;  227     1052  1 ! IMPLICIT OUTPUTS:
;  228     1053  1 !
;  229     1054  1 !       LUB$V_NOECHO which, when set, suppresses terminal echoing.
;  230     1055  1 !
;  231     1056  1 ! ROUTINE VALUE:
;  232     1057  1 ! COMPLETION CODES:
;  233     1058  1 !
;  234     1059  1 !       SS$_NORMAL
;  235     1060  1 !
;  236     1061  1 ! SIDE EFFECTS:
;  237     1062  1 !
;  238     1063  1 !       Signals if an error is encountered.
;  239     1064  1 !       BAS$$CB_PUSH will signal if the channel number is invalid.
;  240     1065  1 !       We signal BAS$K_IO_CHANOT if the channel is not open.
;  241     1066  1 !
;  242     1067  1 !--
;  243     1068  1
;  244     1069  2     BEGIN
;  245     1070  2
;  246     1071  2     BUILTIN
;  247     1072  2         FP;
;  248     1073  2
;  249     1074  2     GLOBAL REGISTER
;  250     1075  2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
;  251     1076  2
;  252     1077  2     LOCAL
;  253     1078  2         FMP : REF BLOCK [, BYTE];
;  254     1079  2
;  255     1080  2     FMP = .FP;
;  256     1081  2 !+
;  257     1082  2 ! Get the CCB for the channel.
;  258     1083  2 !-
;  259     1084  2
;  260     1085  3     IF (.CHAN EQL 0)
;  261     1086  2     THEN
;  262     1087  3         BEGIN
;  263     1088  3 !+
;  264     1089  3 ! The user is referencing his controlling terminal.
;  265     1090  3 !-
;  266     1091  3         BAS$$CB_PUSH (LUB$K_LUN_INPU, LUB$K_ILUN_MIN);
```

```
;  267          1092  3          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  268          1093  !+
;  269          1094  ! If the controlling terminal is not yet open, open it.
;  270          1095  !-
;  271          1096
;  272          1097          IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$OPEN_ZERO (.FMP [SF$L_SAVE_FP]);
;  273          1098
;  274          1099          END
;  275          1100      ELSE
;  276          1101          BEGIN
;  277          1102  !+
;  278          1103  ! This is an ordinary channel.
;  279          1104  !-
;  280          1105          BAS$$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
;  281          1106          CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  282          1107          END;
;  283          1108
;  284          1109  !+
;  285          1110  ! If the channel is not now open, either there is a problem with
;  286          1111  ! the OPEN code, or the non-zero channel was not first opened.
;  287          1112  !-
;  288          1113
;  289          1114      IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  290          1115
;  291          1116  !+
;  292          1117  ! Now set the NOECHO bit, which will cause the record level code
;  293          1118  ! to tell RMS to stop echoing terminal input.
;  294          1119  !-
;  295          1120      CCB [LUB$V_NOECHO] = 1;
;  296          1121  !+
;  297          1122  ! We are done with register CCB.
;  298          1123  !-
;  299          1124      BAS$$CB_POP ();
;  300          1125      RETURN (SS$_NORMAL);
;  301          1126  1   END;                                    ! end of BAS$NOECHO


                                    081C 00000          .ENTRY  BAS$NOECHO, Save R2,R3,R4,R11                ; 1035
                    54 00000000G    00   9E 00002          MOVAB   BAS$$CB_PUSH, R4                             ;
                    53              5D   D0 00009          MOVL    FP, FMP                                      ; 1080
                            04      AC   D5 0000C          TSTL    CHAN                                         ; 1085
                                    1E   12 0000F          BNEQ    1$                                           ;
                    50              08   CE 00011          MNEGL   #8, R0                                       ; 1091
                    52              07   CE 00014          MNEGL   #7, R2                                       ;
                                    64   16 00017          JSB     BAS$$CB_PUSH                                 ;
                FF4C    CB      0C  A3   D0 00019          MOVL    12(FMP), -180(CCB)                           ; 1092
                        29      FC  AB   E8 0001F          BLBS    -4(CCB), 3$                                  ; 1097
                                0C  A3   DD 00023          PUSHL   12(FMP)                                      ;
            00000000G   00              01   FB 00026          CALLS   #1, BAS$$OPEN_ZERO                           ;
                                        0E   11 0002D          BRB     2$                                           ; 1085
                                        50   D4 0002F  1$:     CLRL    R0                                           ; 1105
                        52          04  AC   D0 00031          MOVL    CHAN, R2                                     ;
                                        64   16 00035          JSB     BAS$$CB_PUSH                                 ;
                FF4C    CB          0C  A3   D0 00037          MOVL    12(FMP), -180(CCB)                           ; 1106
```

```
                    0B     FC   AB  E8 0003D 2$:    BLBS    -4(CCB), 3$                                    ; 1114
                    7E     00G  8F  9A 00041       MOVZBL  #BAS$K_IO_CHANOT, -(SP)                        ;
          00000000G 00           01  FB 00045       CALLS   #1, BAS$$STOP_IO                               ;
                AO   AB          01  88 0004C 3$:    BISB2   #1, -96(CCB)                                   ; 1120
                     00000000G   00  16 00050       JSB     BAS$$CB_POP                                    ; 1124
                    50           01  D0 00056       MOVL    #1, R0                                         ; 1125
                                     04 00059       RET                                                    ; 1126

; Routine Size:  90 bytes,    Routine Base:  _BAS$CODE + 005A

;  302          1127  1
;  303          1128  1 END                                         ! end of module BAS$ECHO
;  304          1129  1
;  305          1130  0 ELUDOM



;                                PSECT SUMMARY
;
;      Name                    Bytes                           Attributes
;
;  _BAS$CODE                       180  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                           Library Statistics
;
;                                        --------  Symbols  --------      Pages       Processing
;      File                              Total     Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776          2         0       581          00:01.2



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASECHO/OBJ=OBJ$:BASECHO MSRC$:BASECHO/UPDATE=(ENH$:BASECHO)

; Size:         180 code + 0 data bytes
; Run Time:        00:10.0
; Elapsed Time:    00:26.8
; Lines/CPU Min:    6759
; Lexemes/CPU-Min: 40301
; Memory Used:  119 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASDET
LIS

BASDISPAT
LIS

BASCVTTP
LIS

BASDELETE
LIS

BASDATETI
LIS

BASECHO
LIS

BASCVTRP
LIS

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASENDDFS
LIS

BASERROR
LIS

BASENDDEF
LIS

BASEDIT
LIS

BASEND
LIS

BASEDIVP
LIS

BASEMULP
LIS

BASERRTXT
LIS

BASENDGSB
LIS